Lesson 2: SQL Basics (Part B)

Audience

- Those who are new to relational DBs and SQL.

Objective

To understand basic ***SQL statements***.

To create simple ***Queries: SELECT, DISTINCT, Sorting & Ordering***.

To go beyond simple columns: ***Concatenation, Mathematical, Dates, Null***.

To understand how to ***filter data using the WHERE clause***.

Software & Resources Needed

- Database Server with Interface
  - MS SQL Server with SQL Server Management Studio

  OR

  - MySQL with MySQL Workbench (part of MySQL install)

  OR

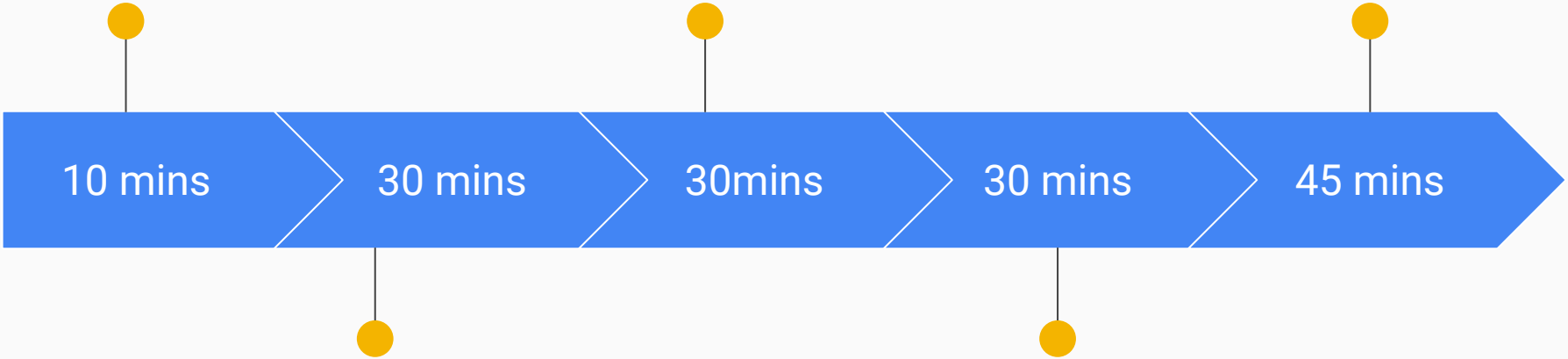
  - Online (sqlfiddle.com)
- Data
  - Test data on Github repo.
- Github repo for information

Overview of Day 2

Basic SQL Queries (SELECT)

Expressions, Casting, NULL

Practice



SELECT, DISTINCT
ORDER BY

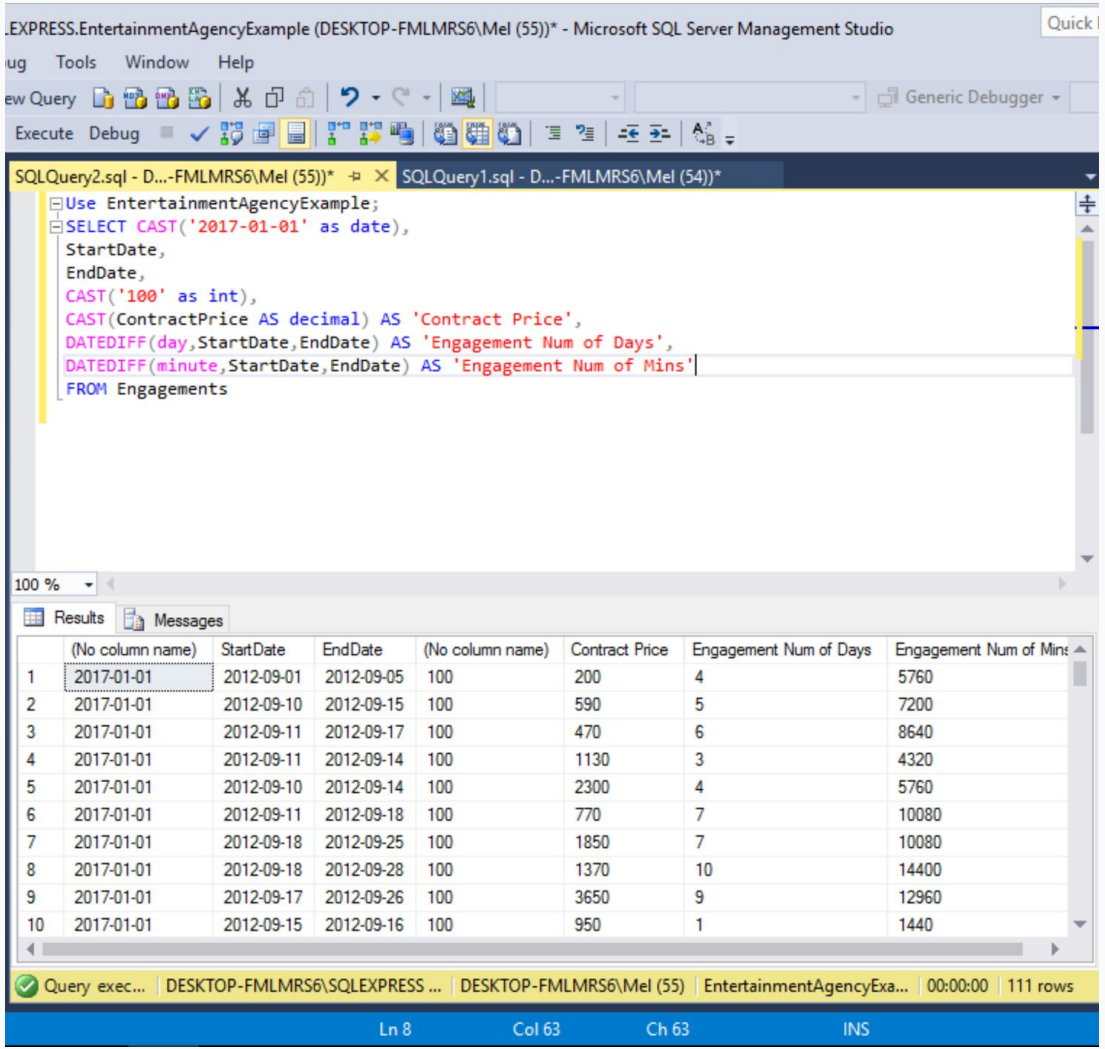
Filtering using WHERE
Clauses

Day 2, Part B: Getting More than Simple Columns

Expressions Overview

Expressions

- To get more than simple columns, you should create an expression.
- Expressions are operations involving numbers, character strings, dates, and times.
- These are functions built into SQL.
- Examples include CAST, AVG, MIN, COUNT, SUM, DATEDIFF, and many more!



Casting

- You can cast one data type into another in a SELECT statement.
- Know that SQL data is typed, just like C# and Java. See http://www.w3schools.com/sql/sql_datatypes_general.asp

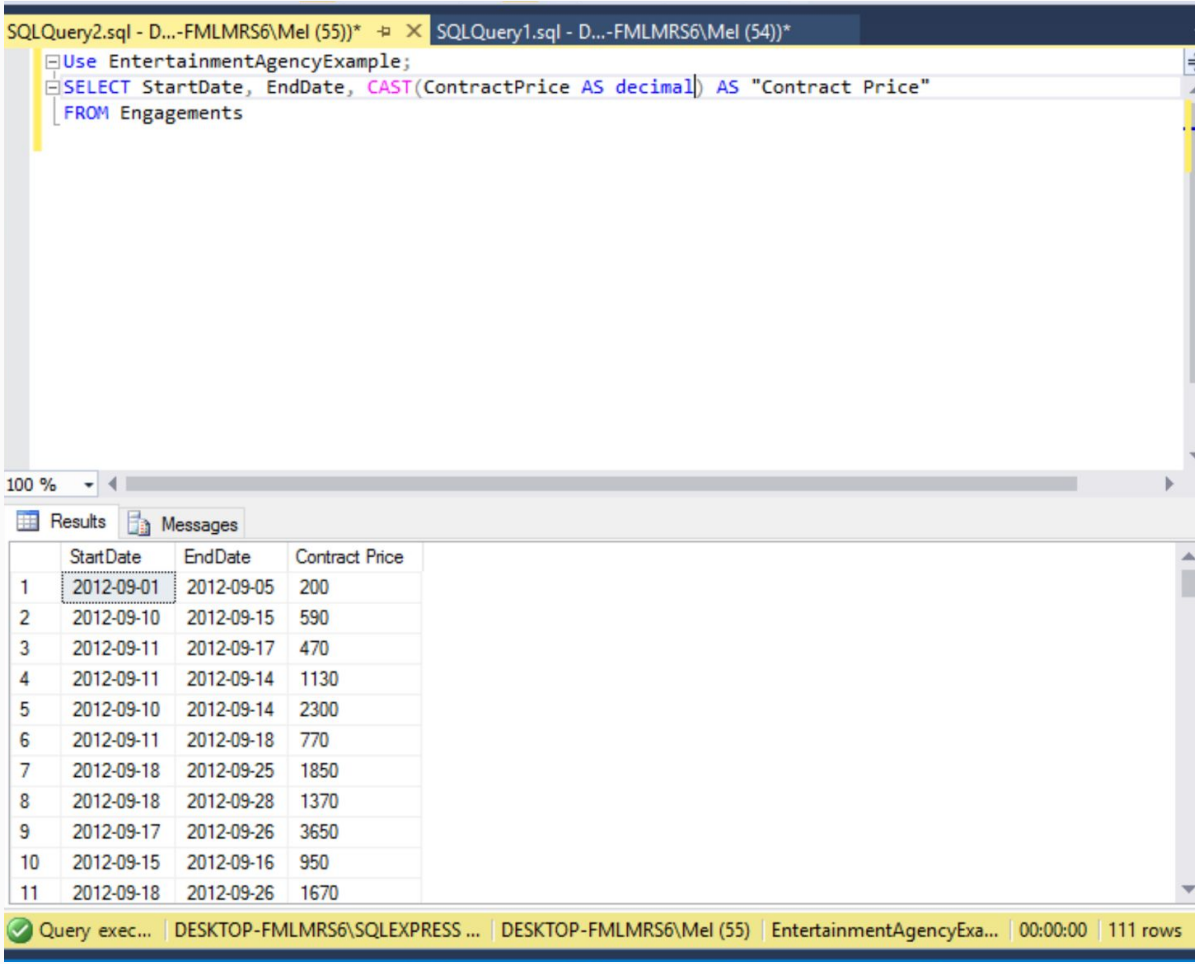
CAST('2016-11-22' AS DATE)

CAST('03:30:25' AS TIME)

CAST('2016-09-29 12:35:00' AS DATETIME)

CAST('100' AS decimal)

Cast(Price as float)



	StartDate	EndDate	Contract Price
1	2012-09-01	2012-09-05	200
2	2012-09-10	2012-09-15	590
3	2012-09-11	2012-09-17	470
4	2012-09-11	2012-09-14	1130
5	2012-09-10	2012-09-14	2300
6	2012-09-11	2012-09-18	770
7	2012-09-18	2012-09-25	1850
8	2012-09-18	2012-09-28	1370
9	2012-09-17	2012-09-26	3650
10	2012-09-15	2012-09-16	950
11	2012-09-18	2012-09-26	1670

Mathematical Expressions

Standard Math Functions

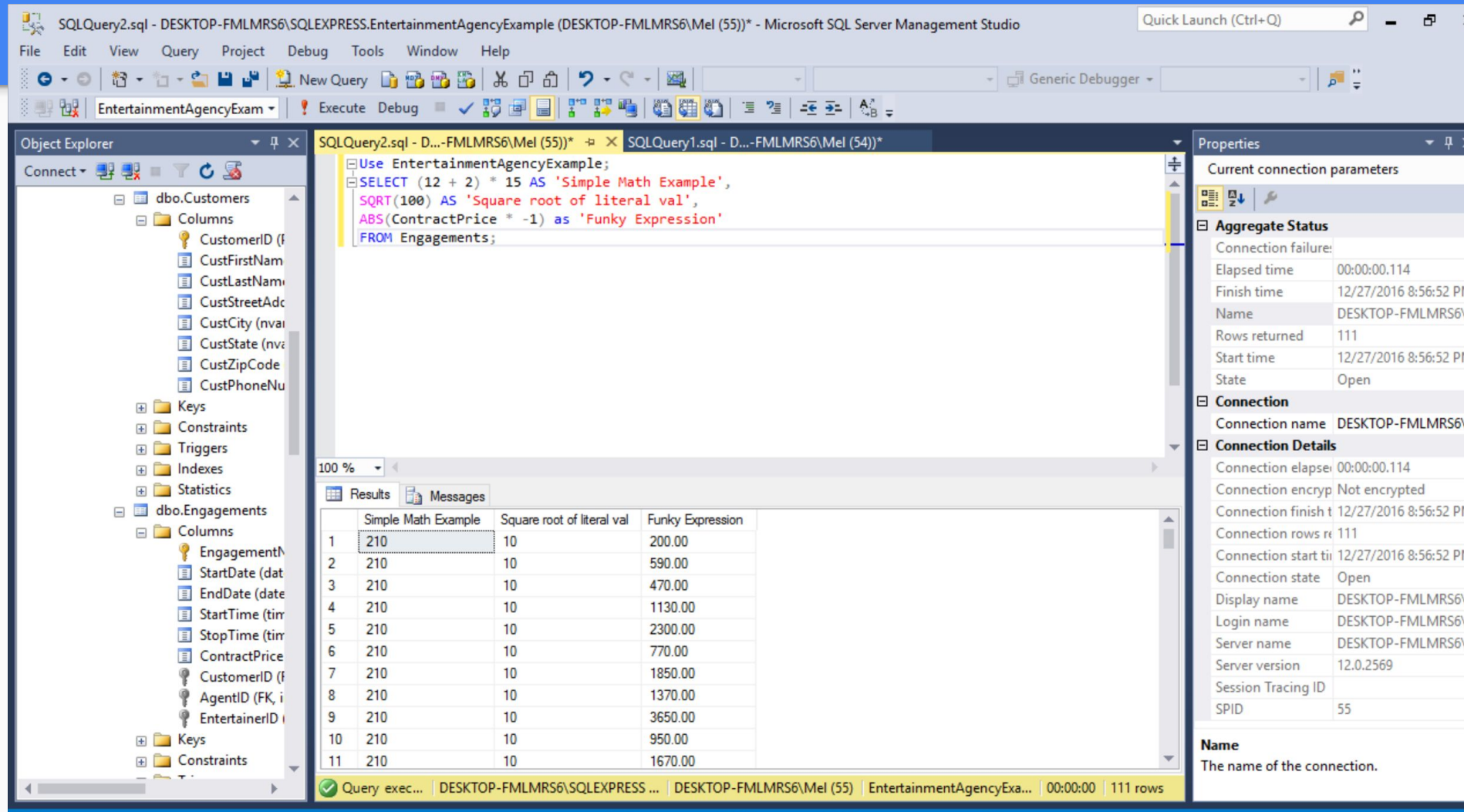
- ABS(numeric expression)
- MOD(dividend, divisor) - doesn't work in MS SQL
- LN(numeric expression)
- EXP(numeric expression)
- POWER(base, exponent)
- SQRT(numeric expression)
- FLOOR(numeric expression)
- CEIL(numeric expression)
  OR
  CEILING(numeric expression)
- WIDTH_BUCKET(numeric value, numeric lower bound, neric upper bound, numeric bucket count)

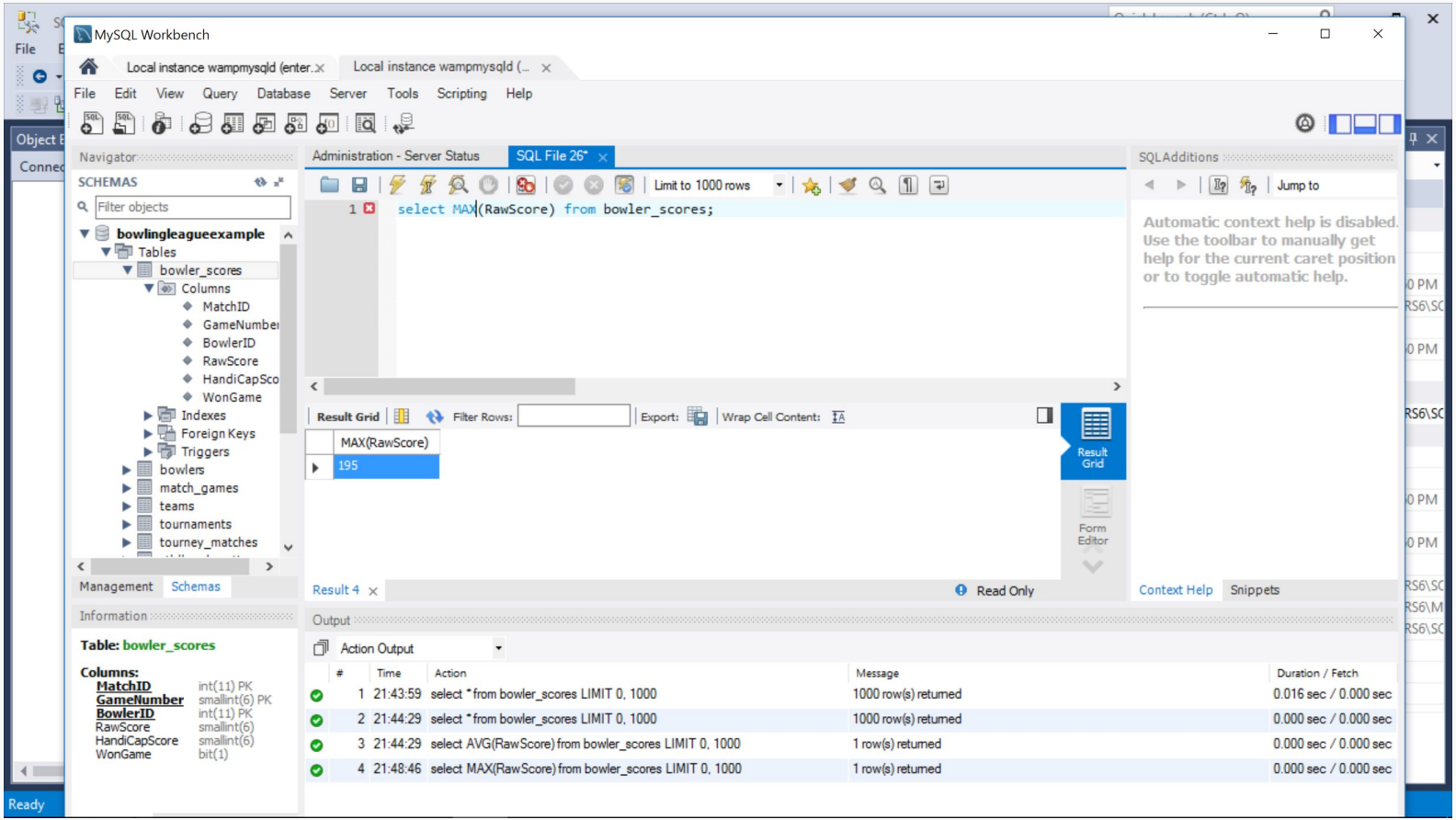
Your job:

- Look up each of these functions to see what they mean.
- Try them out as a SQL query.

NOTES:

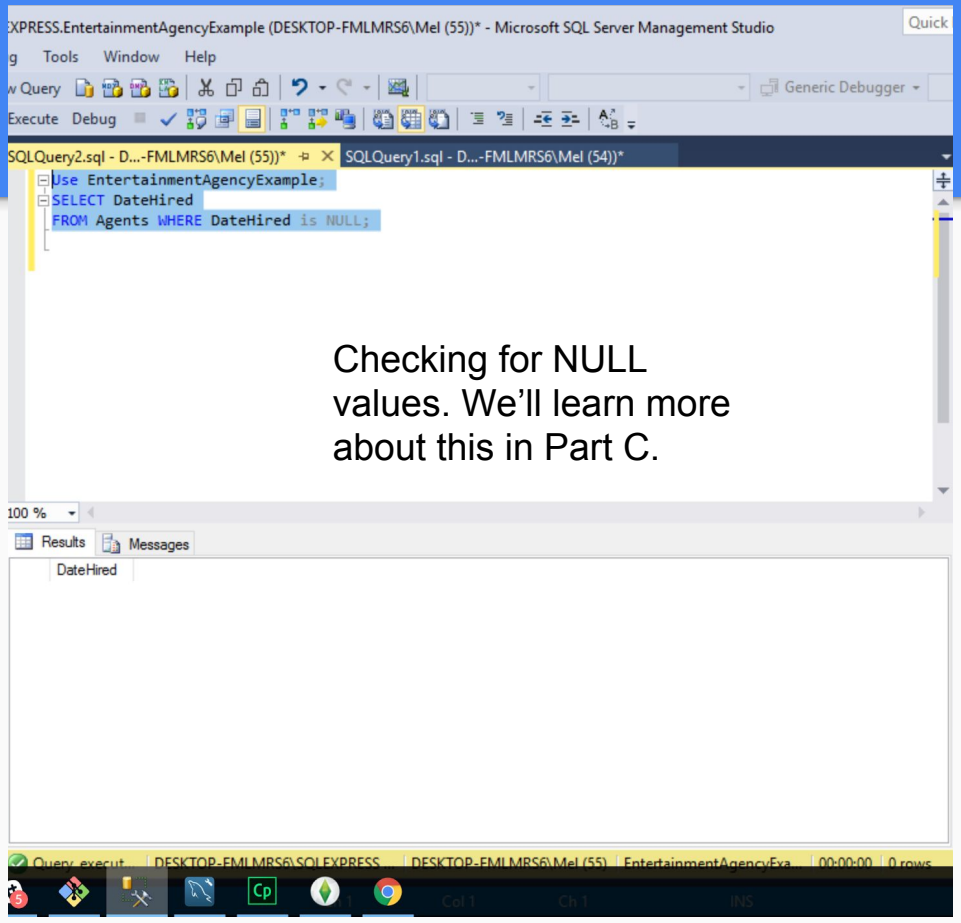
- You can also do standard mathematical expressions, like +, -, *, and /.
- Competing servers (MS SQL vs MySQL) have additional / different expressions.
- Others not listed include AVG(), SUM(), MAX(), and more.


SQLQuery2.sql - DESKTOP-FMLMRS6\SQLEXPRESS.EntertainmentAgencyExample (DESKTOP-FMLMRS6\Mel (55))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Debug Tools Window Help
New Query
Generic Debugger
EntertainmentAgencyExam
Execute Debug
Object Explorer
Connect
dbo.Customers
Columns
CustomerID (
CustFirstNam
CustLastNam
CustStreetAdc
CustCity (nva
CustState (nva
CustZipCode
CustPhoneNu
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Engagements
Columns
EngagementN
StartDate (dat
EndDate (date
StartTime (tim
StopTime (tim
ContractPrice
CustomerID (
AgentID (FK, i
EntertainerID
Keys
Constraints
SQLQuery2.sql - D...-FMLMRS6\Mel (55))*
SQLQuery1.sql - D...-FMLMRS6\Mel (54))*
Use EntertainmentAgencyExample;
SELECT (12 + 2) * 15 AS 'Simple Math Example',
SQRT(100) AS 'Square root of literal val',
ABS(ContractPrice * -1) as 'Funky Expression'
FROM Engagements;
100 %
Results
Messages
Simple Math Example | Square root of literal val | Funky Expression
1 210 10 200.00
2 210 10 590.00
3 210 10 470.00
4 210 10 1130.00
5 210 10 2300.00
6 210 10 770.00
7 210 10 1850.00
8 210 10 1370.00
9 210 10 3650.00
10 210 10 950.00
11 210 10 1670.00
Query exec... DESKTOP-FMLMRS6\SQLEXPRESS ... DESKTOP-FMLMRS6\Mel (55) EntertainmentAgencyExa... 00:00:00 111 rows
Properties
Current connection parameters
Aggregate Status
Connection failures
Elapsed time 00:00:00.114
Finish time 12/27/2016 8:56:52 PM
Name DESKTOP-FMLMRS6\
Rows returned 111
Start time 12/27/2016 8:56:52 PM
State Open
Connection
Connection name DESKTOP-FMLMRS6\
Connection Details
Connection elapse 00:00:00.114
Connection encryp Not encrypted
Connection finish t 12/27/2016 8:56:52 PM
Connection rows re 111
Connection start ti 12/27/2016 8:56:52 PM
Connection state Open
Display name DESKTOP-FMLMRS6\
Login name DESKTOP-FMLMRS6\
Server name DESKTOP-FMLMRS6\
Server version 12.0.2569
Session Tracing ID
SPID 55
Name
The name of the connection.


MySQL Workbench
Local instance wampmysqld (enter...
Local instance wampmysqld (...
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
bowlingleagueexample
Tables
bowler_scores
Columns
MatchID
GameNumber
BowlerID
RawScore
HandiCapSco
WonGame
Indexes
Foreign Keys
Triggers
bowlers
match_games
teams
tournaments
tourney_matches
Management Schemas
Information
Table: bowler_scores
Columns:
MatchID int(11) PK
GameNumber smallint(6) PK
BowlerID int(11) PK
RawScore smallint(6)
HandiCapScore smallint(6)
WonGame bit(1)
Administration - Server Status
SQL File 26*
Limit to 1000 rows
1 select MAX(RawScore) from bowler_scores;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
MAX(RawScore)
195
Result Grid
Form Editor
Result 4
Read Only
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help Snippets
Output
Action Output
Time Action Message Duration / Fetch
1 21:43:59 select * from bowler_scores LIMIT 0, 1000 1000 row(s) returned 0.016 sec / 0.000 sec
2 21:44:29 select * from bowler_scores LIMIT 0, 1000 1000 row(s) returned 0.000 sec / 0.000 sec
3 21:44:29 select AVG(RawScore) from bowler_scores LIMIT 0, 1000 1 row(s) returned 0.000 sec / 0.000 sec
4 21:48:46 select MAX(RawScore) from bowler_scores LIMIT 0, 1000 1 row(s) returned 0.000 sec / 0.000 sec
Ready

NULL values

- Nulls are fields in records that don't contain values.
- Empty strings are different than NULL, as are zeros.
- They exist because of missing values, sometime because the user forgot or didn't know that information when they entered it into the DB.
- If you are having issues, especially with mathematical expressions, NULLS are sometimes the culprits.



Date Expressions

MySQL Date Functions

- NOW()
- CURDATE()
- CURTIME()
- DATE()
- EXTRACT()
- DATE_ADD()
- DATE_SUB()
- DATEDIFF()
- DATE_FORMAT()

Your job:

- Look up each of these functions to see what they mean. http://www.w3schools.com/sql/sql_dates.asp
- Try them out as a SQL query. Do some (all?) work in MS SQL too?

NOTES:

- You can also do standard mathematical expressions on dates, like EndDate - StartDate. I prefer to use DATEDIFF() though.

MS SQL Server Date Functions

- GETDATE()
- DATEPART()
- DATEADD()
- DATEDIFF()
- CONVERT()

Your job:

- Look up each of these functions to see what they mean. http://www.w3schools.com/sql/sql_dates.asp
- Try them out as a SQL query. Do some (all?) work in MySQL too?

NOTES:

- You can also do standard mathematical expressions on dates, like EndDate - StartDate. I prefer to use DATEDIFF() though.

Date Expressions

SQLQuery2.sql - D...-FMLMRS6\Mel (55))* | SQLQuery1.sql - D...-FMLMRS6\Mel (54))*

```
Use EntertainmentAgencyExample;
SELECT
GETDATE() AS [Today's Date],
DATEDIFF(day,StartDate,EndDate) AS 'Engagement length in days'
FROM Engagements;
/*In MySQL, use NOW() instead of GETDATE().*/
```

100 %

Results | Messages

	Today's Date	Engagement length in days
1	2016-12-27 21:14:03.350	4
2	2016-12-27 21:14:03.350	5
3	2016-12-27 21:14:03.350	6
4	2016-12-27 21:14:03.350	3
5	2016-12-27 21:14:03.350	4
6	2016-12-27 21:14:03.350	7
7	2016-12-27 21:14:03.350	7
8	2016-12-27 21:14:03.350	10
9	2016-12-27 21:14:03.350	9
10	2016-12-27 21:14:03.350	1
11	2016-12-27 21:14:03.350	8

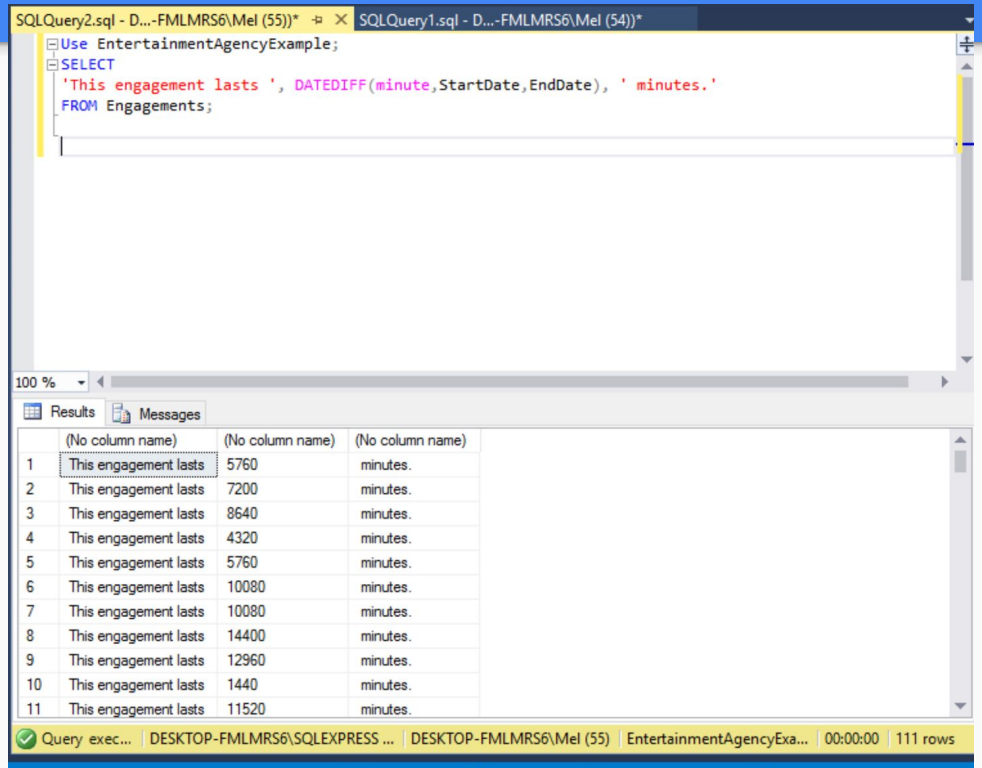
Query exec... | DESKTOP-FMLMRS6\SQLEXPRESS ... | DESKTOP-FMLMRS6\Mel (55) | EntertainmentAgencyExa... | 00:00:00 | 111 rows

Ln 7 Col 1 Ch 1 INS

Concatenation

MySQL Date Functions

- You can concatenate strings In SQL.
- Use string literals as columns in your select.
- Some servers have not implemented the concatenation operator, “II” found in the SQL standard. MS SQL uses “+” or “&.” Commas always work as shown in the image to the right.



Date Expressions

SQLQuery2.sql - D...-FMLMRS6\Mel (55))* | SQLQuery1.sql - D...-FMLMRS6\Mel (54))*

```
Use EntertainmentAgencyExample;
SELECT
GETDATE() AS [Today's Date],
DATEDIFF(day,StartDate,EndDate) AS 'Engagement length in days'
FROM Engagements;
/*In MySQL, use NOW() instead of GETDATE().*/
```

100 %

Results | Messages

	Today's Date	Engagement length in days
1	2016-12-27 21:14:03.350	4
2	2016-12-27 21:14:03.350	5
3	2016-12-27 21:14:03.350	6
4	2016-12-27 21:14:03.350	3
5	2016-12-27 21:14:03.350	4
6	2016-12-27 21:14:03.350	7
7	2016-12-27 21:14:03.350	7
8	2016-12-27 21:14:03.350	10
9	2016-12-27 21:14:03.350	9
10	2016-12-27 21:14:03.350	1
11	2016-12-27 21:14:03.350	8

Query exec... | DESKTOP-FMLMRS6\SQLEXPRESS ... | DESKTOP-FMLMRS6\Mel (55) | EntertainmentAgencyExa... | 00:00:00 | 111 rows

Ln 7 Col 1 Ch 1 INS

Try It

Select the number of seconds for each engagement in the Engagements table located in the EntertainmentAgencyExample database.

Try It

Using the EntertainmentAgencyExample database, concatenate the seconds returned in the previous example so the results state:
“This engagement lasts [your seconds result] seconds.”

Try It

Using the database bowlingleagueexample,
Select the average raw score from all bowler_scores.

Try It

Using the database bowlingleagueexample,
Select the minimum, maximum, and average raw scores from all bowler_scores.

Continued in Day 2, Part C: Filtering Your Data.